Создание карты полета квадрокоптера с подсчетом зафиксированной на ней вражеской техники, и дальнейшим определением её типов с помощью классификатора

Выполнила: Петелина Ярослава Андреевна (11 лет)

Ученица 6 класса ГБОУ г. Москвы №1391

Наставник: Петелина Дарья Сергеевна, kucevol@gmail.com

г. Москва 2022

Цель проекта:

склеить карту из фотографий, полученных с дрона, и посчитать с помощью компьютерного зрения количество вражеской техники.

Результат работы программы

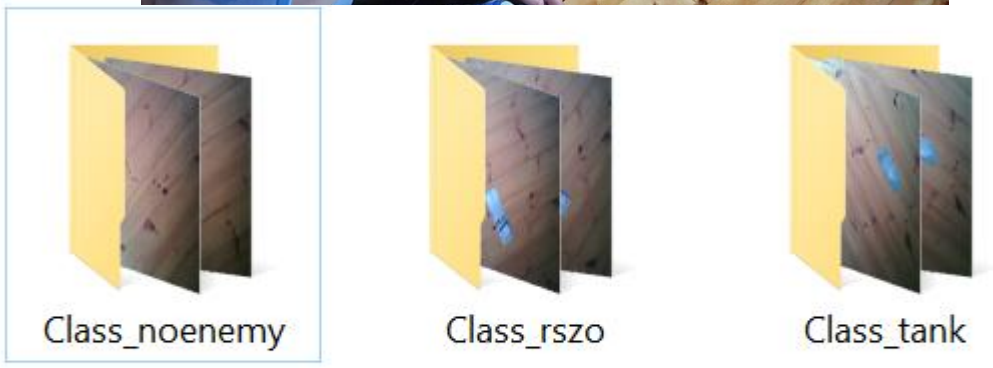
Количество танков: 3

Количество реактивных систем залпового огня: 2

Получение датасета для классификатора

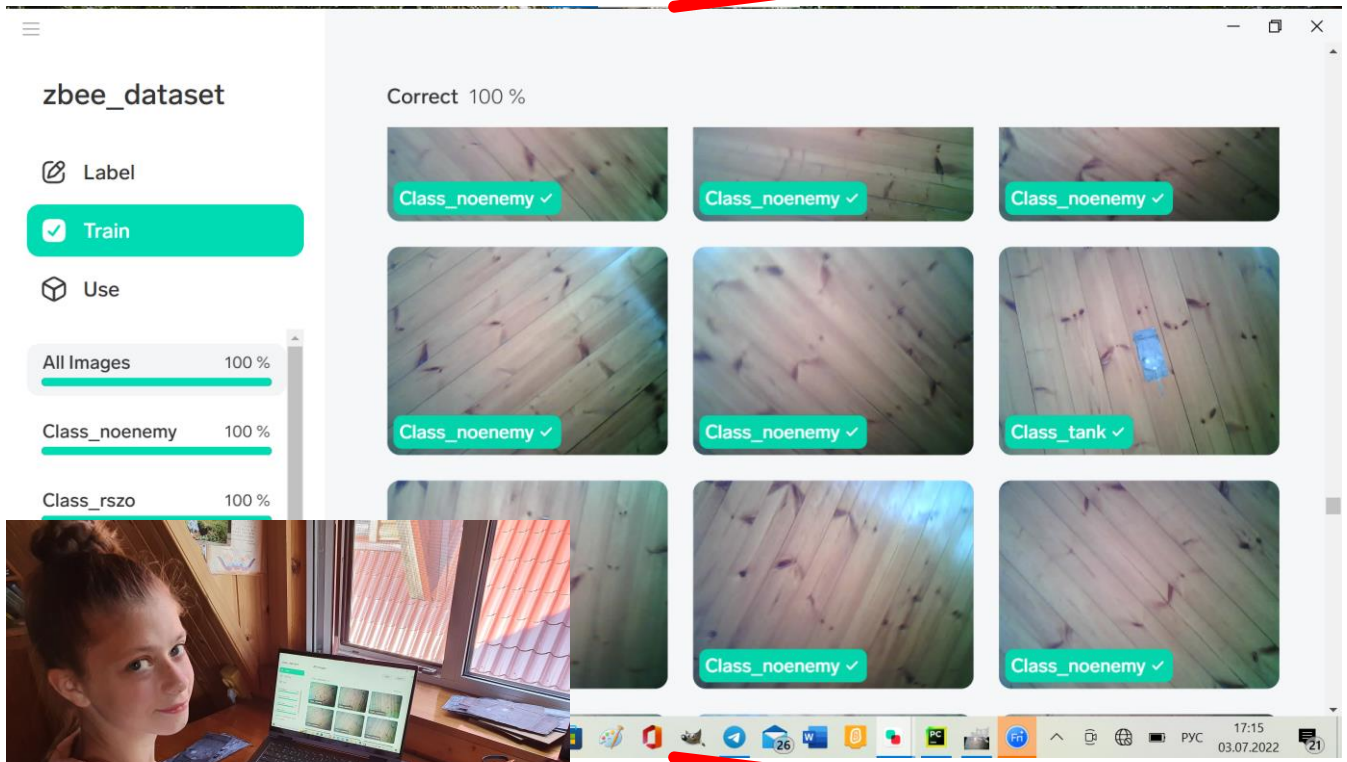
С помощью дрона, очень удобно создавать, почти в автоматическом режиме, датасет. Было решено взять три класса: ничего не обнаружено, РСЗО и танк.

```python
import pioneer_sdk
import cv2
import numpy as np
import os

pioneer = pioneer_sdk.Pioneer()

classes = ('tank', 'rszo', 'noenemy')
indexes = []
cur_class = 0

for cls in classes:
    if f'Class_{cls}' not in os.listdir():
        os.mkdir(f'Class_{cls}')
    indexes.append(len(os.listdir(path=f'./Class_{cls}')))

while True:
    raw = pioneer.get_raw_video_frame()
    frame = cv2.imdecode(np.frombuffer(raw, dtype=np.uint8)

    k = cv2.waitKey(1)

    if k == ord('q'):
        break

    if k == ord('f') and cur_class < len(classes)-1:
        cur_class += 1
    if k == ord('b') and cur_class > 0:
        cur_class -= 1
```



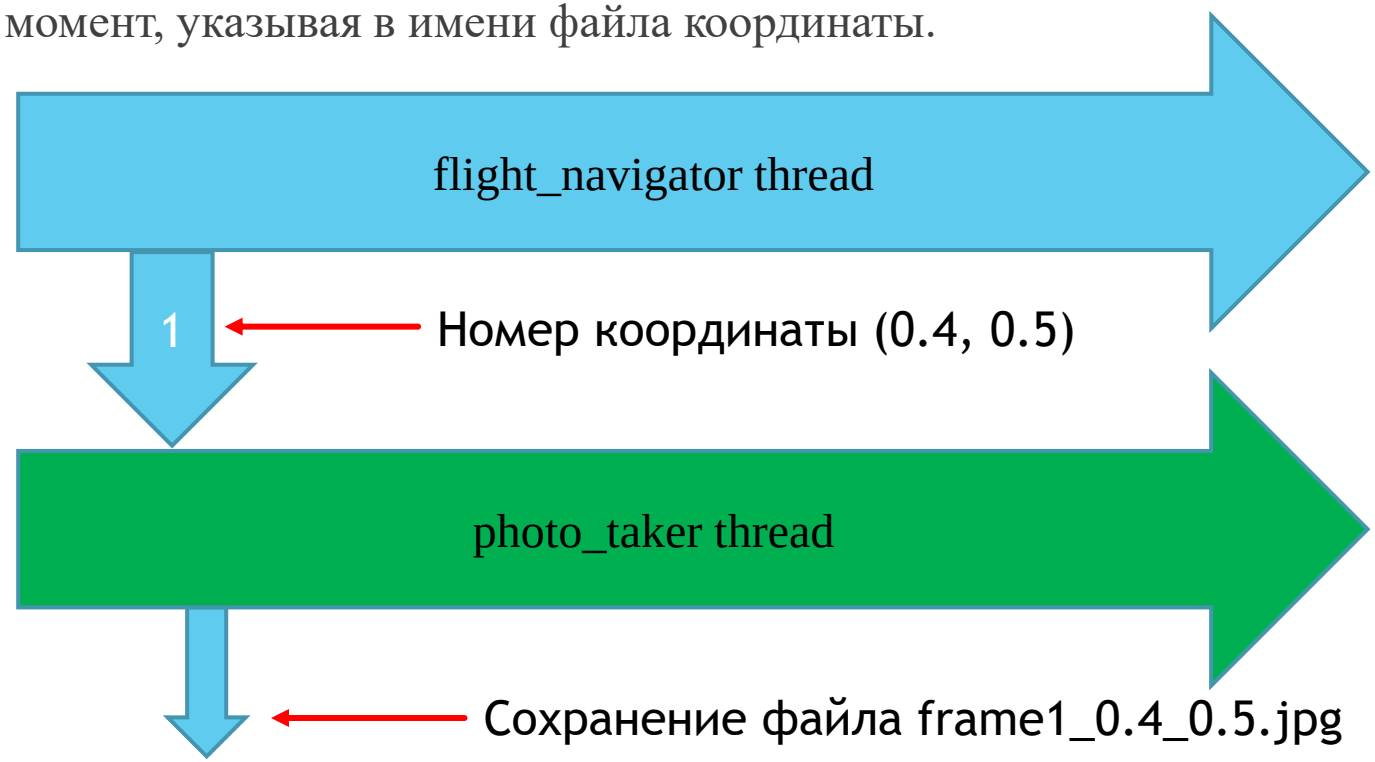
Обучение классификатора

В программе Lobe был обучен классификатор и протестирован на реальных объектах, которые попадали в объектив камеры квадрокоптера. С точностью 99% все объекты распознались правильно.



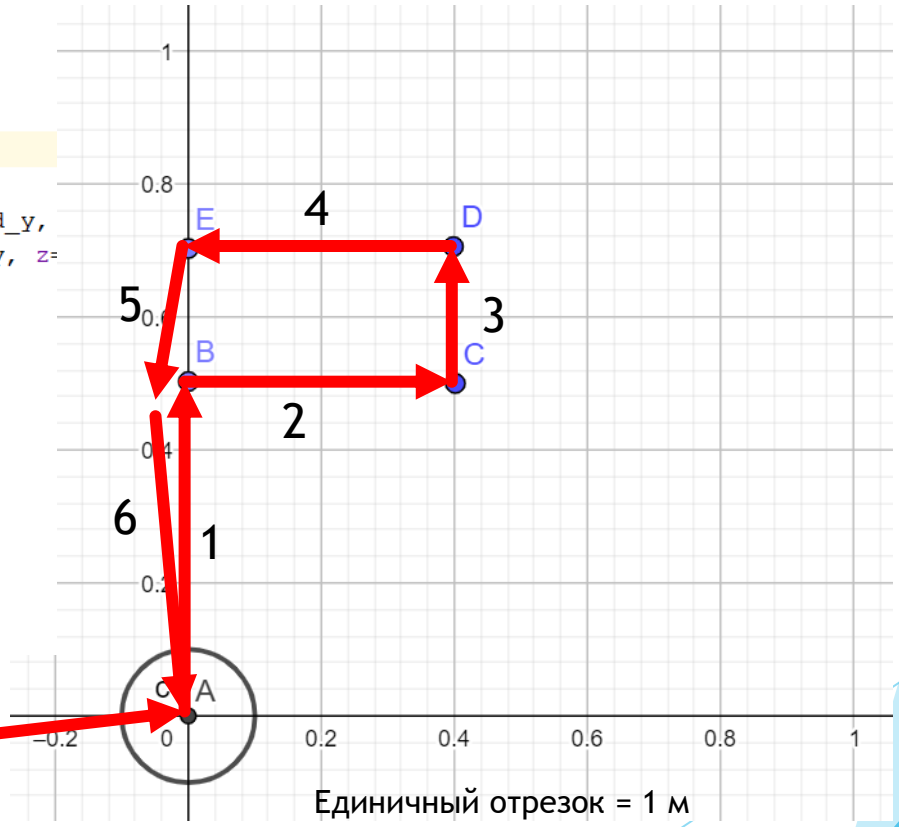
Основная программа

- Программа выполняется в 2 потока: один поток отвечает за полёт по координатам, а другой – за фотографирование и сохранение изображений.
- Координация между потоками происходит с помощью обмена сообщениями: поток, отвечающий за полёт, прибыв в точку, посылает свои координаты второму потоку. Второй поток сохраняет изображение, полученное с камеры дрона в этот момент, указывая в имени файла координаты.



Полёт по координатам

```
i = 0

command_x = x[i]
command_y = y[i]
command_z = float(1)
command_yaw = math.radians(float(0))

while True:
    if new_point:
        print("Летим в точку ", command_x, ", ", command_y,
        drone.go_to_local_point(x=command_x, y=command_y, z=
        new_point = False

    key = cv.waitKey(1)
    if key == 27:  # esc
        print('esc pressed')
        pioneer_mini.land()

        if buff.full():
            buff.get()
        buff.put(['end'])
        break
```



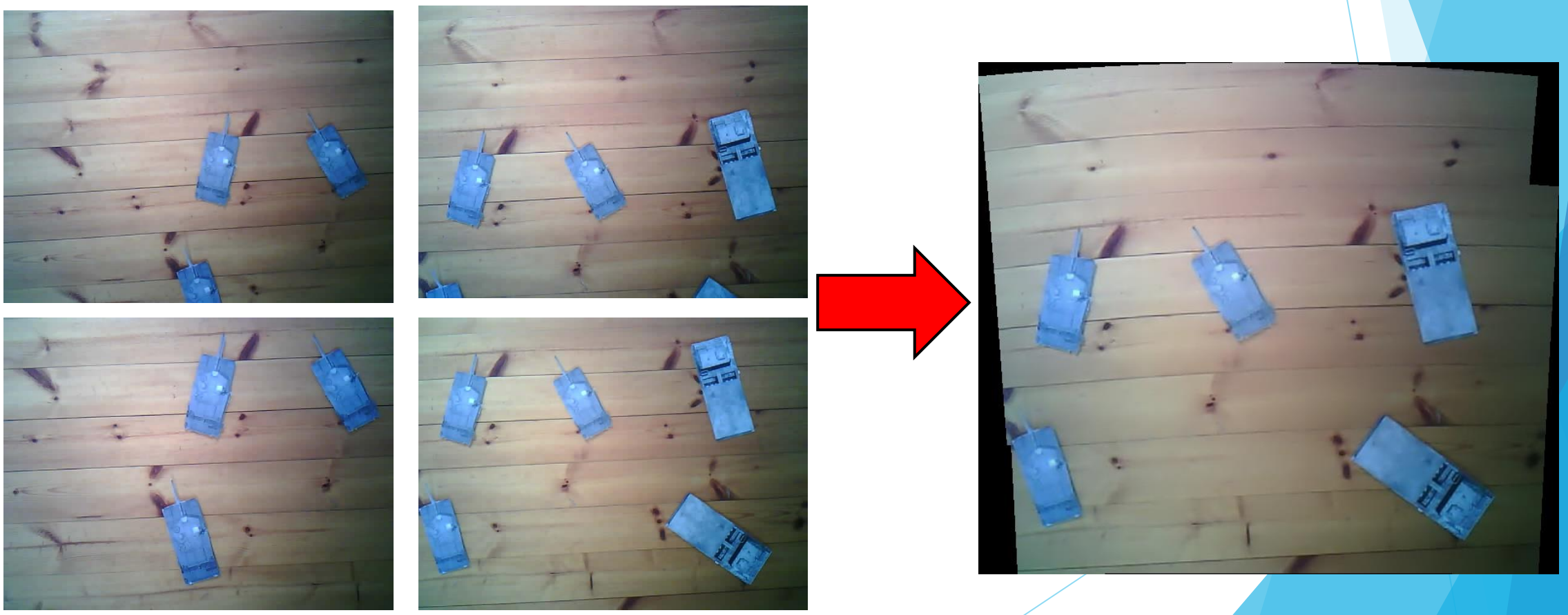
Начало полёта

Единичный отрезок = 1 м

Постобработка фотографий с дрона

Склейка фотографий в единую карту с помощью OpenCV:

```
stitcher = cv.Stitcher.create(cv.Stitcher_PANORAMA)
status, full_map = stitcher.stitch(imgs)
```



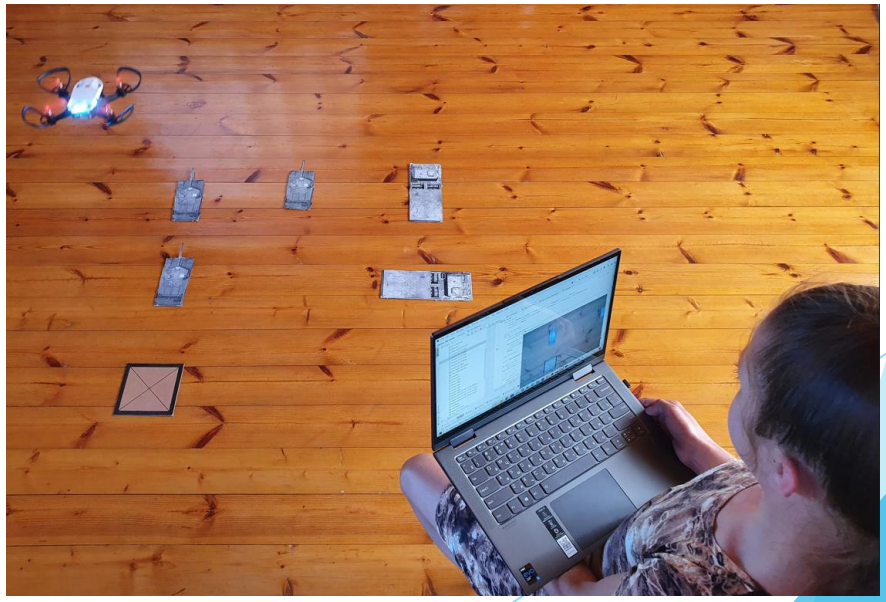
Обработка карты по секторам с помощью классификатора

- Разрезание на секторы с помощью OpenCV, вызов классификатора для каждого кропа и подсчет объектов

```
crop_img = image[y:y+cell_height, x:x+cell_width]
model = ImageModel.load('./zbee_onnx')
predictions = model.predict(model_frame)
```

Предложение про практическому использованию

- Автоматизированная тактическая разведка размещения техники противника с помощью квадракоптера.
- Составление карты расположения вражеских укреплений и войск.
- Получение оператором с безопасного расстояния информации с воздуха.



Планы на будущее

- Заменить разрезание карты на сектора на детектор объектов YOLOv3
- Увеличить точность подсчета объектов на карте
- Усовершенствовать передвижение квадрокоптера по координатам, сейчас столкнулись с неправильной работой функции **point_reached**(blocking=False). При её использовании некоторые координатные точки пропускались, поэтому она была заменена на неэффективный time.sleep().